# Unit 8 Computer architecture and data representation

## Learning objectives

1. Explain how computers use binary numbers to represent numbers and text.
2. Determine the largest integer a computer can store using a given number of bits.
3. Convert numbers from binary to decimal, and from decimal to binary.
4. Count and add numbers in binary.
5. Implement binary-number operations as computer programs (e.g. addition, conversion)
6. Define the terms bit, byte, kilobyte, megabyte, gigabyte and terabyte.
7. Estimate the size in bytes of a typical file type (e.g. a digital movie is about 2 GB, a high-res photo is about 1 MB, a laptop hard drive is about half a terabyte.)
8. Explain the purpose of the ASCII and Unicode encoding systems and describe the differences between them.
9. Explain the differences between the four main types of memory (secondary storage, RAM, virtual memory and registers) in terms of size, speed, permanence of the data, and location within the computer.
10. Identify and describe the functions of the main components on a motherboard, including the CPU, heat sink, northbridge, southbridge, I/O ports, ATX connector, PCI slots, RAM slots and PCI express slot.
11. Explain the function of logic gates, including AND, OR, XOR, NAND and NOT, and predict the output of a circuit of logic-gates from its binary inputs.
12. Explain how binary addition can be implemented as a circuit of logic gates.

|  |  |  |
| --- | --- | --- |
| **Lesson** | **Content** | **Practice/Assessments** |
| **Lesson 1**  Binary numbers | - Light switch demo - How many different states 1, 2, 3 or 4 switches can be in - Computers represent numbers as states of switches (bits) - Counting in binary - Converting binary to decimal - Adding in binary - How computers add numbers:  convert to binary🡪 add in binary 🡪 convert to decimal - Teacher-Tube video: “Binary Numbers” | - Decode binary message  - Write a Python program that converts binary to decimal |
| **Lesson 2**  Binary numbers and text encoding | - Recap binary-to-decimal conversions - Converting decimal to binary - The special binary values 1, 10, 100 etc - Bits, bytes, KB, MB, GB, TB - ASCII encoding, with computer demo  - Unicode encoding, with computer demo | Begin *Computer Memory Assignment.doc* |
| **Lesson 3**  Computer components | - Definition of a computer - Main components of a computer - Investigation: motherboard components | Finish *Computer Memory Assignment* |
| **Lesson 4:** Computer memory | - Recap of motherboard parts - The 4 types of memory: registers, RAM, virtual memory, secondary Storage - Announce in-class exam, and final-project schedule | Python practice: Implement binary-decimal conversion or adding binary numbers |
| **Lesson 5:** Logic gates | - The ALU and the Control Unit - Introduction to Boolean logic - Logic-Gate Investigation - AND, NOT, OR, XOR, NAND | Python practice: Implement logic gates AND, NOT, OR, NAND, NOR, XOR as Python functions |
| **Lesson 6:** Adder circuits | - Review of binary addition - Implementing addition using logic gates - Half-adders - Full adders - Cascading full adders | Python practice: Implement half-adder and full adder |
| **Review** | - Practice Edmodo test |  |
| **Unit Test** |  |  |